1ο επαναληπτικό εργαστήριο

Δεδομένα

- Τα δεδομένα είναι απαραίτητα στοιχεία ενός προγράμματος, καθώς οι βασικές λειτουργίες ενός προγράμματος είναι η επεξεργασία αυτών των δεδομένων και η εξαγωγή αποτελεσμάτων.
- Απαιτείται λοιπόν η δέσμευση κάποιων χώρων μνήμης για να αποθηκευτούν αυτά τα δεδομένα, κατά την διάρκεια της εκτέλεσης του προγράμματος.
- Αυτοί οι χώροι μνήμης που τους χρησιμοποιούμε για την φύλαξη δεδομένων όταν εκτελείται ένα πρόγραμμα, ονομάζονται ανάλογα με την χρήση τους *σταθερές ή μεταβλητές.*

Οι σταθερές

- Οι σταθερές ορίζονται με την οδηγία #define για να τις περιγράψουμε:
- # define PI 3.14159
- Προσέξτε! αυτή η εντολή δεν τελειώνει με το ερωτηματικό ";". Επίσης υπάρχει κενό μεταξύ του PI και του 3.14159.
- Συνήθως τις γράφουμε με κεφαλαία για να τις ξεχωρίζουμε από τις μεταβλητές χωρίς όμως αυτό να είναι δεσμευτικό.

Οι μεταβλητές

- Αντίθετα με τις σταθερές, οι τιμές των μεταβλητών είναι δυνατόν να αλλάξουν κατά την διάρκεια της εκτέλεσης ενός προγράμματος.
- Οι γλώσσες προγραμματισμού μας δίνουν την δυνατότητα να καθορίσουμε μεταβλητές και στην συνέχεια να τους δώσουμε όποια τιμή επιθυμούμε, που έχει βέβαια σχέση με το πρόγραμμα.
- Η δήλωση των μεταβλητών γίνεται συνήθως στην αρχή του προγράμματος.

Ο τύπος char…

- Χρησιμοποιείται για να αποθηκεύσει χαρακτήρες.
- Παραδείγματα δήλωσης (και συνεπώς δέσμευσης μνήμης):
  - char ch1; // Δήλωση
  - char ch3='A'; //Δήλωση & αρχική τιμή

Ο τύπος int…

- Χρησιμοποιείται για τον χειρισμό ακεραίων αριθμών με πρόσημο (..., -2, -1, 0, 1, 2, 3, …) ή χωρίς
- Παραδείγματα δήλωσης:
  - int alpha;
  - short metritis=0, beta;
  - unsigned int arithmos_foititon;
- Παράδειγμα χρήσης:
  - alpha=5;
  - printf("the integer is:%d", alpha); //εμφάνιση στη οθόνη

Ο τύπος float

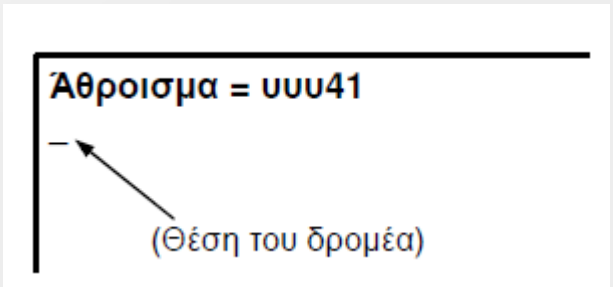
- Χρησιμοποιείται για τον χειρισμό πραγματικών αριθμών (περιλαμβάνουν υποδιαστολή: 34.5, 23.0, 0.987).
- Παραδείγματα δήλωσης:
  - float mesos_oros;
  - float tasi = 5.3;
- Παράδειγμα χρήσης:
  - alpha=12.3;
  - printf("the float is:%f", alpha); //εμφάνιση στη οθόνη
- %lf για πραγματικό αριθμό διπλής ακρίβειας (καταχώριση σε double)

Προσδιοριστές για ακέραιους

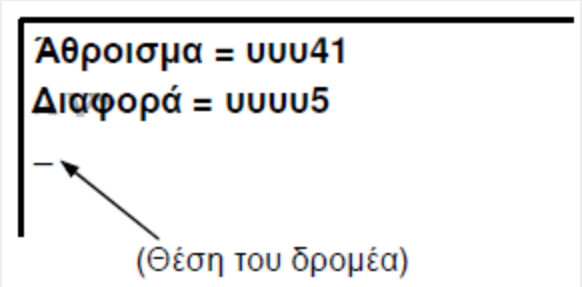
%d : **προσδιοριστής μορφής**. Δηλαδή πού και με ποια μορφή θα εμφανιστεί ένας ακέραιος. Π.χ. το %3d σημαίνει να εμφανιστεί ένας ακέραιος στην οθόνη σε χώρο 3 διαστημάτων.

```
printf (" Άθροισμα = %5d\n ", num+art);
```

Στην πιο πάνω εντολή, αν το num έχει τιμή 23 και το art έχει τιμή 18, στην οθόνη θα εμφανιστεί:

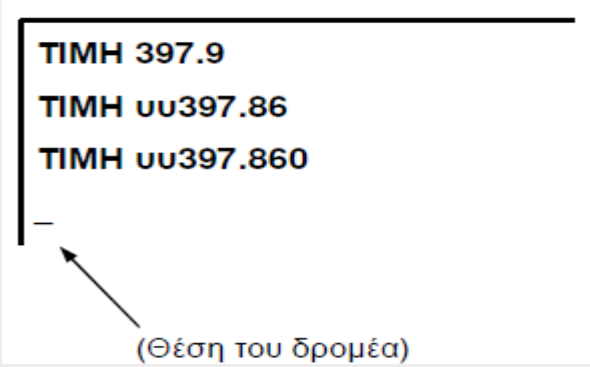



```
printf("Άθροισμα =%d.\n Διαφορά =%5d\n", num+art, num-art);
```



Προσδιοριστές για πραγματικούς

- Float val=397.86;
- printf("ΤΙΜΗ %5.1f\n", val);
- printf("ΤΙΜΗ %8.2\n", val);
- printf("ΤΙΜΗ %9.3\n", val);

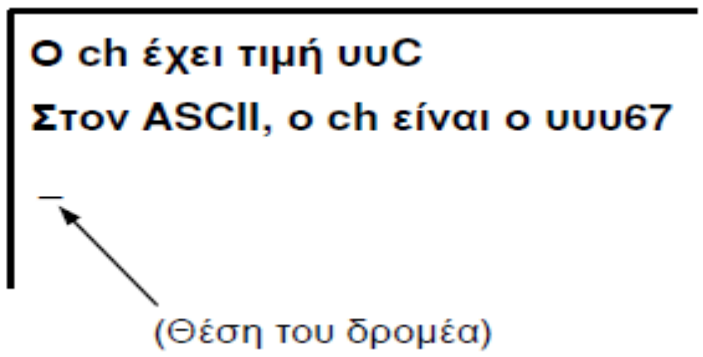


Η printf( ) με προσδιοριστή %4.2f θα εμφανίσει 397.86. Δηλαδή **όταν το ακέραιο μέρος δεν χωράει στο πλήθος των διαστημάτων που καθορίζονται, τότε το ακέραιο μέρος καταλαμβάνει τον χώρο που απαιτείται.**

Προσδιοριστές για χαρακτήρες

Ο %c αποτελεί τον **προσδιοριστή μορφής των χαρακτήρων**. Η printf( ) εμφανίζει τον χαρακτήρα είτε ως ακέραια τιμή είτε ως χαρακτήρα, ανάλογα με τον προσδιοριστή που χρησιμοποιούμε, όπως στο παρακάτω:

```
char ch = 'C';
.....
printf(" Ο ch έχει τιμή %3c\n", ch);
printf(" Στον ASCII ο ch είναι ο %5d\n ", ch);
.....
```



Οι τελεστές στη C

- Η C όπως και κάθε γλώσσα προγραμματισμού χρησιμοποιεί διάφορους τελεστές για να εκτελέσει πράξεις, να αναπαραστήσει αριθμητικές και λογικές εκφράσεις και να δημιουργήσεις δομές ελέγχου.
- Έτσι λοιπόν στην C ανά κατηγορία έχουμε τους τελεστές που ακολουθούν.

Ο τελεστής καταχώρησης:

- Για το τελεστή καταχώρησης χρησιμοποιείται το σύμβολο του ίσον (=) που έχει βέβαια διαφορετική έννοια από ότι στα μαθηματικά. Η εντολή:
- var2 = 52;
- σημαίνει ότι δίνω την τιμή 52 στην μεταβλητή var2. Αν δώσω παρακάτω την εντολή:
- var2 = var2 +1;
- που ενώ μαθηματικά δεν "στέκει", "αυξάνω" κατά μία μονάδα την τιμή που ήδη υπάρχει καταχωρημένη στην μεταβλητή var2, δηλαδή η νέα τιμή της var2 μετά την εκτέλεση της συγκεκριμένης εντολής θα είναι 53.

Οι αριθμητικοί τελεστές:

Πρόκειται για τους:

- + Πρόσθεση - Αφαίρεση * Πολλαπλασιασμός / Διαίρεση
- % Ακέραιο υπόλοιπο (modulo) (πχ., 25 % 4 μας δίνει 1. Εφαρμόζεται μόνο σε ακεραίους)
- Για κάθε τελεστή απαιτούνται 2 όροι (εκτός από τον τελεστή "-", που είναι δυνατόν να χρησιμοποιηθεί και σαν πρόσημο), οι οποίοι μπορεί να είναι σταθερές ή μεταβλητές.
- Ilikia = etos1 - etos2;
- printf("%d", 95-55);
- Celsious = (-25);

- Τα κενά διαστήματα δεν παίζουν ρόλο και μπορείτε να χρησιμοποιήστε παρενθέσεις για μαθηματικές παραστάσεις:
- result = -(b+2)*a + (c +3*(a-b);
- Η C ακολουθεί την αλγεβρική προτεραιότητα στους αριθμητικούς τελεστές.
- Δηλαδή μεγαλύτερη προτεραιότητα έχουν οι παρενθέσεις, μετά το πρόσημο (δηλαδή το μείον), ακολουθούν ο πολλαπλασιασμός, η διαίρεση και το modulo και τέλος οι τελεστές πρόσθεσης και αφαίρεσης.
- Η εντολή καταχώρησης έχει μικρότερη προτεραιότητα από όλες τις πράξεις.
- Η φορά των πράξεων είναι από αριστερά προς δεξιά για όλους τους αριθμητικούς τελεστές, εκτός αν χρησιμοποιήστε το μείον σαν πρόσημο, οπότε η φορά είναι από δεξιά προς αριστερά!

Το πρώτο πρόγραμμα σε C:

```
#include <stdio.h>
main()
{
   printf("Hello world");
}
```

- #include <stdio.h> → Περιλαμβάνει βιβλιοθήκες της C για είσοδο/έξοδο.
- main() → Υποχρεωτική συνάρτηση.
- { → σηματοδοτεί την αρχή των προτάσεων της συνάρτησης main().
- Printf(“ ”) → Εμφανίζει στην οθόνη ότι υπάρχει ανάμεσα στα “” .
- } → σηματοδοτεί την τέλος των προτάσεων της συνάρτησης main().

Η συνάρτηση scanf():

- Η συνάρτηση **scanf()** μας δίνει τη δυνατότητα να αναθέσουμε (δηλαδή να εισάγουμε / διαβάσουμε) τιμές σε μεταβλητές από το πληκτρολόγιο *την ώρα που το πρόγραμμα τρέχει. Η σύνταξή της είναι:*

scanf (ΣΕΙΡΑ_ΕΛΕΓΧΟΥ, **&μεταβλητή_1, &μεταβλητή_2,... , &μεταβλητή_v)**

- όπου **&μεταβλητή_v συμβολίζει τον *δείκτη της μεταβλητής_v***

- Η scanf() αν και μοιάζει πολύ με την printf() έχει μια μεγάλη διαφορά.
- Όπως βλέπετε η σύνταξη της συνάρτησης δεν αναφέρεται σε ονομασίες μεταβλητών, αλλά στους **δείκτες τους.**
- **Δείκτης μιας μεταβλητής είναι ουσιαστικά η διεύθυνσή της στην κύρια μνήμη του υπολογιστή (η διεύθυνση του χώρου που καταλαμβάνει η μεταβλητή).**
- **Στην γλώσσα προγραμματισμού C, για ν' αναφερθούμε στην διεύθυνση μιας μεταβλητής χρησιμοποιούμε το όνομα της μεταβλητής προσθέτοντας μπροστά το σύμβολο "&".**

Δείτε και ένα ολοκληρωμένο πρόγραμμα:

```
 #include <stdio.h>
#include <stdlib.h>
main()  {
int index;
float num;
printf (“Give me the integer index: ” );
scanf(“%d”, &index); //διαβάζω από το πληκτρολόγιο τη τιμή της index
printf (“ Give me the real (float) num: ”);
scanf("%f", &num); //διαβάζω από το πληκτρολόγιο τη τιμή της num
printf(“The value of index is %d.\n”, index); //εμφανίζω τη τιμή της
    index
printf(“The value of num is %f.\n”, num); //εμφανίζω τη τιμή της num
num = 27.567; //αλλάζω τη τιμή της num
printf("The value of num is %f. \n”, num); //εμφανίζω τη τιμή της num
    –έχει αλλάξει
system("PAUSE");
}
```

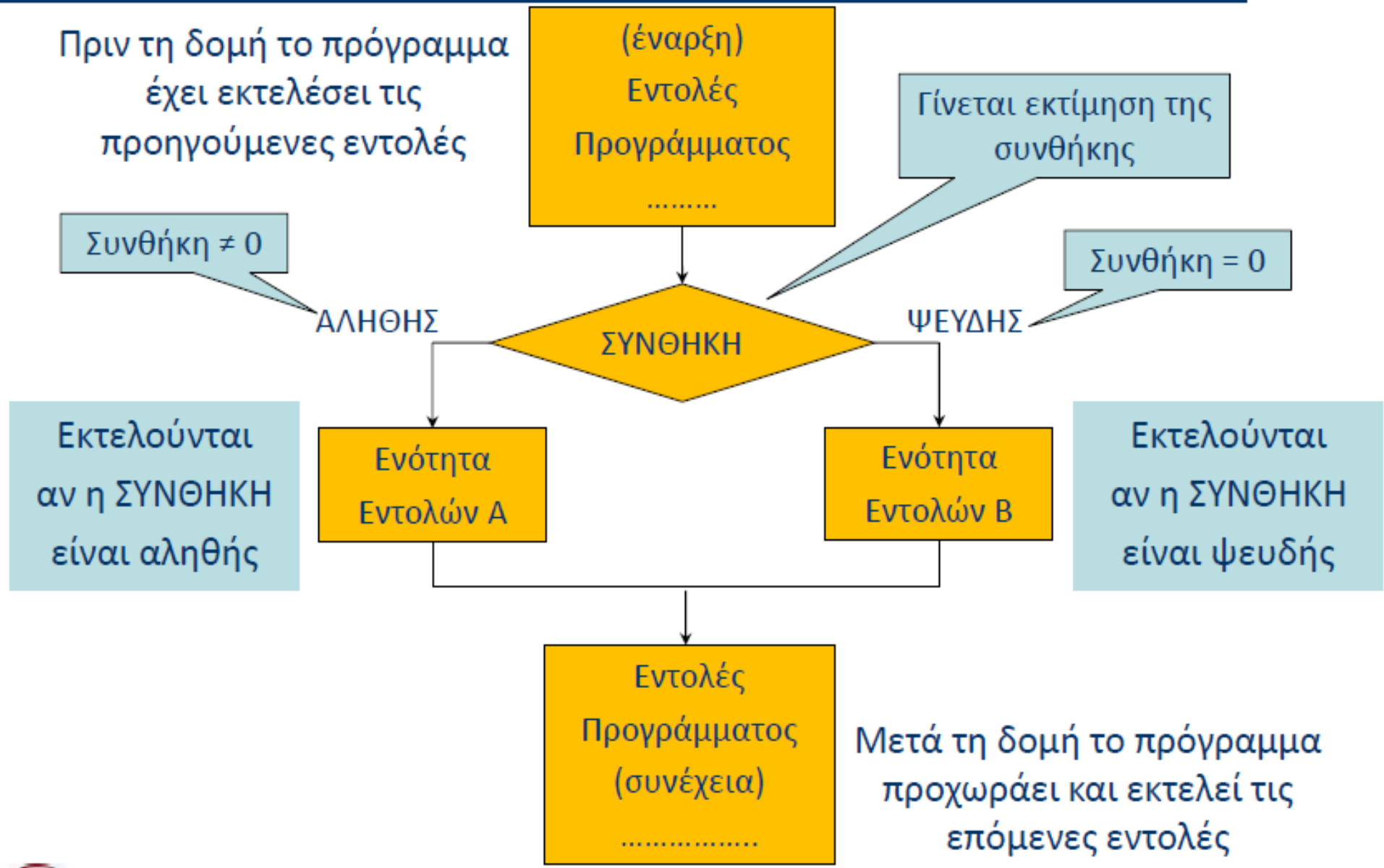
```
 Give me the integer index: 5
 Give me the real (float) num: 8.77
The value of index is 5
The value of num is 8.770000
The value of num is 27.566999
Press any key to continue . . .


--------------------------------
Process exited after 14.35 seconds with return value 0
Press any key to continue . . .
```

Η εντολή if/else

- Οι εντολές if και if/else υπάρχουν σχεδόν σε όλες τις γλώσσες προγραμματισμού.
- Χρησιμοποιούνται για να εκτρέψουν την ροή εκτέλεσης ενός προγράμματος ανάλογα με το αν ισχύει ή όχι κάποια συνθήκη.
- Στην C η γενική σύνταξη της εντολής είναι:

```
if (<ΣΥΝΘΗΚΗ>)
   Ενότητα εντολών-Α
else
   Ενότητα εντολών -Β
```

Γραφική αναπαράσταση της πλήρους δομής



```
if (<ΣΥΝΘΗΚΗ>)
   Ενότητα εντολών-Α
else
   Ενότητα εντολών -Β
```

- Η <ΣΥΝΘΗΚΗ> μπορεί να είναι έκφραση συσχετισμού, λογική πρόταση, αποτέλεσμα κάποιας πράξης, είτε ακόμα και κάποια μεταβλητή ή τιμή.
- Οι ενότητες εντολών Α και Β μπορεί να αποτελούνται από μια ή περισσότερες εντολές (block εντολών) που περικλείονται σε άγκιστρα (**{ , }**).
- **Αν κάποια ενότητα αποτελείται από μια μόνο εντολή τότε τα άγκιστρα είναι προαιρετικά, οπότε συνήθως παραλείπονται.**

Οι τελεστές συσχετισμού

- Πρόκειται για τελεστές που τους χρησιμοποιούμε για την δημιουργία απλών λογικών εκφράσεων συσχετισμού (σύγκρισης).

<	Μικρότερο από
>	Μεγαλύτερο από
<=	Μικρότερο από ή ίσο
>=	Μεγαλύτερο από ή ίσο
==	Λογικό ίσον
!=	Διάφορο (όχι ίσο)

- Το αποτέλεσμα μιας λογικής έκφρασης είναι είτε **1** (αληθές) είτε **0** (ψευδές).
- Οι τελεστές σύγκρισης χρησιμοποιούνται σαν συνθήκες κυρίως στις εντολές ελέγχου και επανάληψης.

Λογικοί τελεστές

- Πρόκειται για τελεστές που μας επιτρέπουν να συνδυάσουμε απλές εκφράσεις συσχετισμού και να δημιουργήσουμε πιο πολύπλοκες λογικές προτάσεις.

&&	Λογικό "και" (AND)
||	Λογικό διαζευτικό "ή" (είτε - OR)
!	Λογική άρνηση (NOT)

- Το αποτέλεσμα μιας πρότασης με λογικούς τελεστές είναι είτε **μηδέν** (0 -ψευδές), είτε **όχι μηδέν** (αληθές).
- Με τους τελεστές && και ||, απαιτείται η χρήση 2 όρων, ενώ με τον τελεστή ! ενός (μοναδιαίος).

```
x > 10 && x < 20

(x > 10 && x < 20) || x > 50
```

```
if (<ΣΥΝΘΗΚΗ>)
   Ενότητα εντολών-Α
else
   Ενότητα εντολών -Β
```

- Όταν η εκτέλεση του προγράμματος φτάσει στην εντολή **if, τότε επαληθεύεται (εξετάζεται) η <ΣΥΝΘΗΚΗ>.**
- Αν μεν η συνθήκη είναι **αληθής (τιμή συνθήκης διάφορη του 0) τότε εκτελούνται οι εντολές (ή η εντολή) της Ενότητας-Α.**
- Αν πάλι η συνθήκη είναι **ψευδής (τιμή συνθήκης μηδέν), τότε εκτελούνται οι εντολές (ή η εντολή) της Ενότητας-Β**
- Αν η συνθήκη είναι ψευδής και δεν υπάρχει **else (ούτε φυσικά Ενότητα–Β) τότε η εντολή if δεν κάνει τίποτα και το πρόγραμμα προχωρά στην εκτέλεση των εντολών που ακολουθούν.**

Παραδείγματα χρήσης:

(α)

```
if (a == 0) {
        printf("O akeraios %d einai MHDEN\n", a);
        a++;
}
```

(β)

```
if (a == 0) {
        printf("O akeraios %d einai MHDEN\n", a);
        a++;
 }
 else printf("O akeraios %d den einai
   MHDEN\n", a);
```

(γ) **Πολλαπλό if-else.**

Στην περίπτωση που μια από τις δύο (ή και οι δύο) ενότητες εντολών ενός if-else είναι επίσης εντολές if-else τότε έχουμε ένα πολλαπλό if-else. Πχ:

```
if (a >= 0) {
      if (b >= 0){
          printf( "Ta a kai b einai kai ta dyo THETIKA H
          MHDEN\n);
      }
      else {
          printf("To a einai THETIKO H MHDEN alla to b
          einai ARNHTIKO\n);
      }
else {
      printf("To a einai ARNHTIKO\n);
}
```

- Στο παράδειγμα αυτό, επειδή όλες οι ενότητες εντολών αποτελούνται από μια εντολή η καθεμιά, θα μπορούσαμε να παραλείψουμε τελείως όλα τα άγκιστρα. Τότε πώς θα ξέραμε σε ποιο if αντιστοιχεί το κάθε else; Εδώ είναι χρήσιμος ο παρακάτω κανόνας.
- **ΚΑΝΟΝΑΣ:** ***Σε ένα πολλαπλό if-else, κάθε else αναφέρεται στο (ταιριάζει με το) πλησιέστερο πριν από αυτό if που δεν έχει ήδη ταιριάξει με κάποιο άλλο else.***
- Έτσι στο παραπάνω παράδειγμα, το πρώτο else που συναντάμε ταιριάζει με το **if(b>0) ενώ το δεύτερο else λίγο παρακάτω ταιριάζει με το if(a>0) (Παρόλο που το if(b>0) είναι κοντινότερο στο δεύτερο else, δεν μας κάνει, διότι το έχουμε ήδη ταιριάξει με το δικό του else).**

ΠΡΟΣΟΧΗ!

- Δεν βάζουμε το ελληνικό ερωτηματικό «;» **στο τέλος της *if ή της else εντολής.***

- Προσοχή ο τελεστής ελέγχου ισότητας == **(διπλό ίσον) είναι διαφορετικός από τον τελεστή εκχώρησης = (μονό ίσον).**

(δ) Και ένα ολοκληρωμένο πρόγραμμα:

```c
#include <stdio.h>
#include <stdlib.h>
main()
  {    // Άγκιστρο που σηματοδοτεί την αρχή του προγράμματος
     int A,B;
     printf("Enter A: ");
     scanf("%d", &A); // Διαβάζει από το πληκτρολόγιο την τιμής της ακέραιας μεταβλητής A
     printf("Enter B: ");
     scanf("%d", &B); // Διαβάζει από το πληκτρολόγιο την τιμής της ακέραιας μεταβλητής B
     if (A<=0)    // Αρχή εντολής if-else. Έλεγχος συνθήκης
     {            // Το A είναι αρνητικό ή μηδέν. Εκτελούνται οι εντολές της Ενότητας-A
        A+=B;     // 1η εντολή Ενότητας Εντολών-A. Τι κάνει ο τελεστής +=  ;
        B--;      // 2η εντολή Ενότητας Εντολών-A. Τι κάνει ο τελεστής --  ;
     }            // Άγκιστρο τέλους της 1ης ενότητας εντολών του if/else
     else
     {            // Το A είναι θετικό  (> 0). Εκτελούνται οι εντολές της Ενότητας-B
        A-=B;     // 1η εντολή Ενότητας Εντολών-B. Τι κάνει ο τελεστής -=  ;
        B++;      // 2η εντολή Ενότητας Εντολών-B. Τι κάνει ο τελεστής ++  ;
     }            // Άγκιστρο τέλους της  2ης ενότητας εντολών του if/else
     printf("A=%d, B:%d\n", A,B);  // εμφάνιση των νέων τιμών που έχουν τα A και B
  }    // Άγκιστρο που σηματοδοτεί το τέλος του προγράμματος
```

*Η εντολή **switch***: Χρησιμοποιείται εναλλακτικά στη θέση ενός πολλαπλού if-else.

```
switch (έκφραση)
 {
   case σταθερά_1:
      Ενότητα εντολών 1;      // εκτελούνται αν έκφραση = σταθερά_1
     break;

   case σταθερά_2:
      Ενότητα εντολών 2;      // εκτελούνται αν έκφραση = σταθερά_2
     break;
    . . .
   default:                   // Η χρήση του default είναι προαιρετική. Η ενότητα εκτελείται
      Ενότητα εντολών ;       // αν η έκφραση δεν ισούται με καμία από τις παραπάνω σταθερές
       break;
 }
```

Το τελευταίο κομμάτι της εντολής (ενότητα **default**) είναι προαιρετικό και μπαίνει μόνο όταν θέλουμε να γίνει κάτι στην περίπτωση που η έκφραση είναι διαφορετική απ' όλες προηγούμενες σταθερές. Δείτε το επόμενο παράδειγμα:

Κώδικας με ***πολλαπλό if/else***	Ο ίδιος κώδικας με ***switch***

```
if (Day==1)
   printf ("Monday\n");
else if (Day == 2)
   printf ("Tuesday \n");
else if (Day == 3)
   printf ("Wednesday \n");
else if (Day == 4)
   printf ("Thursday \n");
else if (Day == 5)
   printf ("Friday \n");
else if (Day==6 || Day==7)
   printf ("Weekend \n");
else
  printf("Invalid data \n");
```

```
switch (Day) {
  case 1:
     printf ("Monday\n");
     break;
  case 2:
     printf ("Tuesday \n");
     break;
  case 3:
     printf ("Wednesday \n");
     break;
  case 4:
     printf ("Thursday \n");
     break;
  case 5:
     printf ("Friday \n");
     break;
  case 6:
  case 7:
     printf ("Weekend \n");
     break;
  default
     printf ("Invalid data \n");
     break;
}
```

Παρατηρήστε στο παράδειγμα ότι μπορούμε να αντιστοιχίσουμε περισσότερες από μια σταθερές στην την ίδια ενότητα εντολών.

```c
// Μετατροπή από ίντσες σε εκατοστά, ή το αντίστροφο
#include <stdio.h>

int main ()
{
   const double cm_per_inch = 2.54;                // πλήθος cm ανά inch
   int length;                                     // απόσταση σε ίντσες ή εκατοστά
   char unit;                                      // μονάδα μέτρησης, i:ίντσες ή c:εκατοστά

   printf ("Παρακαλώ εισάγετε μία απόσταση και τη μονάδα της: ");
   scanf ("%d %c", length, unit);

   switch (unit) {
      case 'i':
            printf ("%din = %fcm\n", length, cm_per_inch * length );
            break;
      case 'c':
            printf ("%dcm = %fin\n", length, length / cm_per_inch);
            break;
      default:
            printf ("Λυπούμαι, δε γνωρίζω μονάδα με το σύμβολο %c\n", unit);
   }
}
```

Επιλογή-switch -Τεχνικές λεπτομέρειες

- η τιμή την οποία πρέπει να επιλέξουμε πρέπει να είναι **int ή char**
- οι τιμές στο case πρέπει να είναι σταθερές εκφράσεις
- δε χρησιμοποιούμε '**;' στο τέλος των case και default, αλλά ':'**
- η τιμή που επιλέγουμε συγκρίνεται διαδοχικά με τις σταθερές των case
- αν η τιμή ταιριάζει με τη σταθερά ενός case τότε εκτελούνται οι εντολές του
- αν η τιμή δεν ταιριάζει με καμία σταθερά, τότε εκτελούνται οι εντολές του default

– εάν δεν υπάρχει το default ⇨ τερματίζεται η εντολή switch και εκτελούνται οι εντολές μετά το switch

- Δε μπορούμε να χρησιμοποιήσουμε την ίδια τιμή για δύο case.
- η εντολή break προκαλεί τον τερματισμό μιας εντολής switch
- η εκτέλεση του προγράμματος μέσα σε ένα switch συνεχίζεται στο επόμενο case, εκτός αν δεν μεσολαβεί κάποια από τις εντολές break
- μπορούμε να αντιστοιχούμε πολλαπλά case σε ένα σύνολο εντολών
  - παραλείπουμε τις αντίστοιχες εντολές break
- ο μεταγλωττιστής δε θα μας προειδοποιήσει εάν ξεχάσουμε κάποιο break

tickets.cpp

Γράψτε ένα πρόγραμμα το οποίο να μιμείται την λειτουργία ενός μηχανήματος αυτόματης πώλησης εισιτηρίων. Στο μηχάνημα πωλούνται 3 κατηγορίες εισιτηρίων:

- Κατηγορία 1. Πολυτέκνων: 0.70 €
- Κατηγορία 2. Φοιτητικό: 0.80 €
- Κατηγορία 3. Ολόκληρο: 1.20 €

Το πρόγραμμα θα ζητάει από το χρήστη την **κατηγορία και το πλήθος των εισιτηρίων (υποθέστε ότι όλα είναι της ίδιας κατηγορίας), καθώς και το ποσό των χρημάτων που δίνει, και θα εμφανίζει το κόστος των εισιτηρίων και τα ρέστα.**
Στην περίπτωση που τα χρήματα που έδωσε ο χρήστης δεν φτάνουν, θα πρέπει αντί για ρέστα να εμφανίζει το υπολειπόμενο ποσό.

```c
#include <stdio.h>
#include <stdlib.h>

main()
{
 int category, plithos;
 float kostos,poso_pliromis, teliko_poso;
  printf("Dose katigoria eisitirion:\n");
  scanf("%d", &category);
  printf("Dose plithos eisitirion:\n");
  scanf("%d", &plithos);
  printf("Dose poso pliromis:\n");
  scanf("%f", &poso_pliromis);

switch (category){

   case 1:
    kostos=0.70*plithos;
    break;
    case 2:
    kostos=0.80*plithos;
    break;
    case 3:
    kostos=1.20*plithos;
    break;
   }
teliko_poso=poso_pliromis-kostos;
if (teliko_poso>0)
    printf("Ta %d eisitiria kostizoun %.2f euro.\nExeis resta %.2f euro.\n", plithos, kostos,teliko_poso);
else
    printf("Ta %d eisitiria kostizoun %.2f euro.\nPrepei na doseis akoma %.2f euro.\n",plithos, kostos, kostos-poso_pliromis);
system("PAUSE");  /* pause */
}
```

Παραδείγματα λειτουργίας του προγράμματος – με **κόκκινο** τα δεδομένα από το πληκτρολόγιο

Dose katigoria eisitirion: 2	Dose katigoria eisitirion: 3
Dose plithos eisitirion: 4	Dose plithos eisitirion: 2
Dose poso pliromis: 5.00	Dose poso pliromis: 2.00
Ta 4 eisitiria kostizoun **3.20** euro	Ta 2 eisitiria kostizoun **2.40** euro
Exeis resta **1.80** euro	Prepei na doseis akoma **0.40** euro

```c
#include <stdio.h>
main()
{
int a,b;
printf("Dwse 2 arithmous: ");
scanf("%d %d",&a,&b);
if (a>b)
if (a>10)
printf("To a einai megalytero tou b kai megalytero tou 10");
else
printf("To a einai megalytero tou b kai mikrotero h iso tou 10");
else
if (b==10)
printf("To a den einai megalytero tou b kai to b einai 10");
else
printf("To a den einai megalytero tou b kai to b den einai 10");
return 0;
}
```

```
Dwse 2 arithmous: 5 9
To a den einai megalytero tou b kai to b den einai 10
--------------------------------
Process exited after 19.19 seconds with return value 0
Press any key to continue . . .
```

```c
#include <stdio.h>
main()
{
int a;
printf("dose akeraio: ");
scanf("%d",&a);
if (a==5)
printf("a=5\n");
else if (a==8)
printf("a=8\n");
else if (a>20)
printf("megalo\n");
else
printf("telos programmatos\n");
return 0;
}
```

```
dose akeraio: 51
megalo

--------------------------------
Process exited after 13.98 seconds with return value 0
Press any key to continue . . .
```